Multi-Bernoulli based Track-Before-Detect with Road Constraints

Jeffrey Wong, Ba Tuong Vo, Ba Ngu Vo
School of Electrical, Electronic and Computer Engineering
University of Western Australia
Crawley, WA 6009
Email: jeffwong@ieee.org, {ba-tuong.vo, ba-ngu.vo}@uwa.edu.au

Reza Hoseinnezhad
School of Aerospace, Mechanical and Manufacturing Engineering
RMIT University
Bundoora, VIC 3083
Email: rezah@rmit.edu.au

***Abstract*—The random set based multi-Bernoulli filter is applied to a challenging low signal to noise track before detect scenario. Specifically we use the variant of the multi-Bernoulli filter that processes raw image observations. We add an additional layer of track management logic to output trajectories rather than point estimates. The tracker also exploits additional road map information by integrating the roads into the filtering likelihood. We show that this approach of using the image observation MeMBer filter with track management and road constrained model can yield an effective tracker for track before detect scenarios.**

I. Introduction

Tracking of dim targets is a challenging prospect especially if the level of background noise is high. Traditional tracking algorithms are designed to use point measurements typically generated by thresholding the sensors output [1]. Thresholding noisy sensor information results in many false detections. A solution to this problem is to use an alternative approach which uses the whole sensor output as a measurement. These algorithms are referred to as track-before-detect (TBD) algorithms.

Salmond and Birch established the first Bayesian TBD algorithm based on a particle implementation [3]. There now exists many new TBD algorithms such as the one by Czyz et. al. which jointly detects and tracks color targets in video [4], the non Bayesian TBD algorithm by Buzzi et. al. [5] and the Histogram Probabilistic Multi-Hypotheis tracker by Streit et. al. [6]. A comprehensive summary of TBD algorithms can be found in Davey's performance comparison [7].

The low intensity of the target return maybe inherent in the particular sensing technology that is used. Sensors which have a low dynamic range like monochrome cameras would be a sensor technology would be affected greater. This would make tracking of aerial imagery particularly difficult especially when combined with having to estimate multiple time varying objects.

A recent approach to tracking is to represent the multi-object state as a finite set and model the multi-target uncertainty by a random finite set. The center piece of the FISST approach to target tracking is the multi-target Bayes filter. Unfortunately it is extremely expensive to run and only lately has there been a discovery which makes it tractable [9]. Due to the inherent combinatorial nature of multi-object densities and the multiple integrations on the (infinite dimensional) multi-object state and observation spaces, the multi-target Bayes filter is intractable in most practical applications. To alleviate this intractability, the probability hypothesis density (PHD) and subsequently Cardinalized PHD (CPHD) filters have been proposed as moment [10] and cardinality approximations [11]. These filters operate on the single-object state space and avoid the combinatorial number of calculations that arises from data association. Since their inceptions the PHD and CPHD filters have generated substantial interest from academia as well as the commercial sector with the developments of numerical implementations such as the sequential Monte Carlo (SMC) or particle [12] and Gaussian mixture[13] PHD filter as well as the CPHD filter [14]. In addition to the PHD and CPHD filters, the multi-target multi-Bernoulli (MeMBer) filter is another approximation of the multi-target Bayes filter, based on multi-Bernoulli RFS [15].

In this paper we consider the problem of tracking multiple ground targets that are contrained to move a net work of roads from image observations. We solve this problem efficiently by using the Image Observation-MeMBer (IO-MeMBer) filter [16] and exploiting the prior terrain information by constraining motion to known roads. This allows for a better localization of our targets. Moreover, we incorporated track management to the output of the IO-MeMBer filter so that output is not purely location estimates but target trajectories.

The paper is organized as follows. Background on the IO-MeMBer filter is presented in Section II. Section III describes how we incorporate prior terrain information and integrate a track management scheme with the IO-MeMBer filter. A numerical simulation is presented in Section IV. We conclude with closing remarks in Section V.

II. The multi-Bernoulli filter using image observations

The multi-Bernoulli filter for point measurements was originally proposed by Mahler [8] and was subsequently shown to have a significant bias which was corrected by cardinality balancing in [15]. A multi-Bernoulli filter for image measurements also referred as to as a TBD filter was later proposed in [16]. The Multi-Bernoulli TBD filter requires the asssumption of a separable likelihood whereby the noise and target distributions can be treated independently. A special case of separability in the multi-object likelihood occurs when objects in the image observation do not overlap or interact with each other. This form of image observations admits an analytical data update [16], [17], [18]. A full description of the derivation from first principles including probablity generating functionals can be found in [16]. This section will describe the Multi-Bernoulli RFS, the IO-MeMBer recursion and its sequential Monte-Carlo implementation.

A. The Multi-Bernoulli RFS

This subsection will briefly describe the definition of a multi-Bernoulli RFS.

A Bernoulli RFS X on a given state space $\mathbb{X}$ is described by two parameters: its existence probability r and its probability density p (defined on χ). The salient feature of a Bernoulli RFS is that any realization takes on either the empty set (with probability $1-r$) or a singleton (with probability r and distributed according to p). The probability density for a Bernoulli RFS is given by (see [8] pp. 368):

$$\pi\left(X\right)=\begin{cases}1-r & X=\emptyset\\ r\centerdot p\left(x\right) & X=\{x\}\end{cases}$$

A multi-Bernoulli RFS X on a given state space $\mathbb{X}$ is the union of a finite number of independent Bernoulli RFSs:

$$X=\cup_{i=1}^{M}X^{(i)}$$

where $X^{(i)}$ are M independent Bernoulli RFSs with existence probabilities $r^{(i)}$ and probability densities $p^{(i)}$. A multi-Bernoulli RFS is thus completely described by the multi-Bernoulli parameter set $\{(r^{(i)},p^{(i)})\}_{i=1}^{M}$. Moreover, its probability density is given by (see [8] pp. 368)

$$\pi(\{x_1,...,x_n\})=\prod_{j=1}^{M}\left(1-r^{(j)}\right)\sum_{1\leq i_1\neq\ldots,\neq i_n\leq M}\prod_{j=1}^{n}\frac{r^{(i_j)}p^{(i_j)}(x_j)}{1-r^{(i_j)}}.$$

The multi-Bernoulli RFS is used in the IO-MeMBer filter to capture the multi-object state of the system and propagate it internally to update it with new information.

B. The IO-MeMBer filter

A multi-object likelihood is said to be separable where the observation vector or image y conditioned on the multi-object state X is given by

$$g\left(y|X\right)=f\left(y\right)\prod_{x\in X}g_y\left(x\right),$$

where $f\left(\cdot\right)$ is probability density independent of X, and $g_y\left(\cdot\right)$ is a probability density parameterized by y. An important special case of a separable likelihood is that of rigid objects observed in background noise. Suppose we have a finite set of object states X. For each $x\in X$, the object at state x illuminates a set of pixels denoted by $T(x)$. A pixel with index i which is illuminated due to an object presence has value distributed according to $\varphi_i(\cdot,x)$. A pixel with index i which is illuminated due to background noise has value distributed according to $\phi_i(\cdot)$. Thus the probability density of the value of pixel y_i given a state x is

$$p(y_i|x)=\left\{\begin{array}{cc}\varphi_i(y_i,x) & i\in T(x)\\ \phi_i(y_i) & i\notin T(x)\end{array}\right..$$

In this case the multi-object likelihood is

$$g\left(y|X\right)=\left[\prod_{x\in X}\prod_{i\in T(x)}p(y_i|x)\right]\left[\prod_{i\notin\cup_{x\in X}T(x)}p(y_i|x)\right],$$

which is of separable form with

$$\begin{aligned}g_y\left(x\right)&=\prod_{i\in T(x)}\frac{\varphi_i(y_i,x)}{\phi_i(y_i)},\\ f\left(y\right)&=\prod_{i}\phi_i(y_i).\end{aligned}$$

Assuming likelihood separability, the posterior density of the *multi-Bernoulli* multi-object RFS admits a closed form recursion under the Chapman-Kologorov equation and the Bayes update equation, i.e. the multi-Bernoulli RFS density admits an exact prediction and update no approximations such cardinality balancing. The recursion is provided here:

Prediction: Given the posterior multi-Bernoulli components $\pi_{k-1}=\{(r_{k-1}^{(i)},p_{k-1}^{(i)})\}_{i=1}^{M_{k-1}}$, where $r_{k-1}^{(i)}$ is the existence probability and $p_{k-1}^{(i)}$ is the object location PDF of target i at time $k-1$, the predicted multi-Bernoulli components are:

$$\pi_{k|k-1}=\{(r_{P,k|k-1}^{(i)},p_{P,k|k-1}^{(i)})\}_{i=1}^{M_{k-1}}\cup\{(r_{\Gamma,k}^{(i)},p_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}},$$

where the existence probability and object location PDF are:

$$\begin{aligned}r_{P,k|k-1}^{(i)}&=r_{k-1}^{(i)}\langle p_{k-1}^{(i)},p_{S,k}\rangle,\\ p_{P,k|k-1}^{(i)}(x)&=\frac{\langle f_{k|k-1}(x|\cdot),p_{k-1}^{(i)}p_{S,k}\rangle}{\langle p_{k-1}^{(i)},p_{S,k}\rangle},\end{aligned}$$

and also :

$f_{k|k-1}(\cdot|\zeta)$ is the *single target transition density,*
$p_{S,k}(\zeta)$ is the *probability of target existence,*
$\{(r_{\Gamma,k}^{(i)},p_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}}$ are the *Bernoulli components,* of the birth RFS at time k.

Update: Given the predicted multi-Bernoulli components $\pi_{k|k-1}=\{(r_{k|k-1}^{(i)},p_{k|k-1}^{(i)}\}_{i=1}^{M_{k|k-1}}$,where $r_{k|k-1}^{(i)}$ is the existence probability and $p_{k|k-1}^{(i)}$ is the object location PDF of target i at time k , the updated multi-Bernoulli components are:

$$\pi_k = \{(r_k^{(i)}, p_k^{(i)})\}_{i=1}^{M_{k|k-1}} \quad (1)$$

where each component is re-weighted using the multi-object likelihood g_y:

$$\begin{aligned} r_k^{(i)} &= \frac{r_{k|k-1}^{(i)} \left\langle p_{k|k-1}^{(i)}, g_y \right\rangle}{1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \left\langle p_{k|k-1}^{(i)}, g_y \right\rangle}, \\ p_k^{(i)} &= \frac{p_{k|k-1}^{(i)} g_y}{\left\langle p_{k|k-1}^{(i)}, g_y \right\rangle}. \end{aligned}$$

Merging: A heuristic method is used to account for the assumption of no overlap. Estimates within a threshold T_{merge} have their existence probabilities $r_k^{(i)}$ and location densities $p_k^{(i)}$ combined together.

State extraction: The same state extraction process as [15] can be used to extract estimates from the multi-Bernoulli components. The expected number of objects or maximum a posteriori cardinality can be estimated. State estimates can be extracted by then selecting that number of states with the highest existence probabilities.

C. SMC implementation of the IO-MeMBer filter

In this section we show the SMC implementation of the prediction(taken from [15]) and for the new update (1) step.

SMC Prediction: Suppose for time equal $k-1$, the (multi-Bernoulli) posterior multi-object density $\pi_{k-1} = \{(r_{k-1}^{(i)}, p_{k-1}^{(i)})\}_{i=1}^{M_{k-1}}$ is given and each $p_{k-1}^{(i)}$, $i = 1, ..., M_{k-1}$, is comprised of a set of weighted particles $\{w_{k-1}^{(i,j)}, x_{k-1}^{(i,j)}\}_{j=1}^{L_{k-1}^{(i)}}$, i.e.

$$p_{k-1}^{(i)}(x) = \sum_{j=1}^{L_{k-1}^{(i)}} w_{k-1}^{(i,j)} \delta_{x_{k-1}^{(i,j)}}(x).$$

Then, given the following proposal densities $q_k^{(i)}(\cdot|x_{k-1}, y_k)$ and $b_k^{(i)}(\cdot|y_k)$, the predicted (multi-Bernoulli) multi-object density can be calculated as follows

$$\begin{aligned} r_{P,k|k-1}^{(i)} &= r_{k-1}^{(i)} \sum_{j=1}^{L_{k-1}^{(i)}} w_{k-1}^{(i,j)} p_{S,k}(x_{k-1}^{(i,j)}), \\ p_{P,k|k-1}^{(i)}(x) &= \sum_{j=1}^{L_{k-1}^{(i)}} \tilde{w}_{P,k|k-1}^{(i,j)} \delta_{x_{P,k|k-1}^{(i,j)}}(x), \\ r_{\Gamma,k}^{(i)} &= \textit{parameter given by birth model}, \\ p_{\Gamma,k}^{(i)}(x) &= \sum_{j=1}^{L_{\Gamma,k}^{(i)}} \tilde{w}_{\Gamma,k}^{(i,j)} \delta_{x_{\Gamma,k}^{(i,j)}}(x), \end{aligned}$$

where

$$\begin{aligned} x_{P,k|k-1}^{(i,j)} &\sim q_k^{(i)}(\cdot|x_{k-1}^{(i,j)}, y_k), \ j = 1, ..., L_{k-1}^{(i)} \\ w_{P,k|k-1}^{(i,j)} &= \frac{w_{k-1}^{(i,j)} f_{k|k-1}(x_{P,k|k-1}^{(i,j)}|x_{k-1}^{(i,j)}) p_{S,k}(x_{k-1}^{(i,j)})}{q_k^{(i)}(x_{P,k|k-1}^{(i,j)}|x_{k-1}^{(i,j)}, y_k)}, \\ \tilde{w}_{P,k|k-1}^{(i,j)} &= w_{P,k|k-1}^{(i,j)} / \sum_{j=1}^{L_{k-1}^{(i)}} w_{P,k|k-1}^{(i,j)} \\ x_{\Gamma,k}^{(i,j)} &\sim b_k^{(i)}(\cdot|y_k) \ j = 1, ..., L_{\Gamma,k}^{(i)} \\ w_{\Gamma,k}^{(i,j)} &= \frac{p_{\Gamma,k}(x_{\Gamma,k}^{(i,j)})}{b_k^{(i)}(x_{\Gamma,k}^{(i,j)}|y_k)}, \\ \tilde{w}_{\Gamma,k}^{(i,j)} &= w_{\Gamma,k}^{(i,j)} / \sum_{j=1}^{L_{\Gamma,k}^{(i)}} w_{\Gamma,k}^{(i,j)}. \end{aligned}$$

SMC Update: Suppose that at time k, the predicted (multi-Bernoulli) multi-object density $\pi_{k|k-1} = \{(r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)})\}_{i=1}^{M_{k|k-1}}$ is given and each $p_{k|k-1}^{(i)}$, $i = 1, ..., M_{k|k-1}$, is comprised of a set of weighted particles $\{w_{k|k-1}^{(i,j)}, x_{k|k-1}^{(i,j)}\}_{j=1}^{L_{k-1}^{(i)}}$, i.e.

$$p_{k|k-1}^{(i)} = \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} \delta_{x_{k|k-1}^{(i,j)}}(x).$$

Then, the updated (multi-Bernoulli) multi-object density (1) calculated as follows

$$\begin{aligned} r_k^{(i)} &= \frac{r_{k|k-1}^{(i)} \varrho_k^{(i)}}{1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \varrho_k^{(i)}} \\ p_k^{(i)} &= \frac{1}{\varrho_k^{(i)}} \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} g_{y_k}(x_{k|k-1}^{(i,j)}) \delta_{x_{k|k-1}^{(i,j)}}(x), \end{aligned}$$

where $\varrho_k^{(i)} = \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} g_{y_k}(x_{k|k-1}^{(i,j)})$.

Re-sampling should be implemented in the same method as in [15] and [16].

III. Adding Road Constraints and Track Management

Our approach in this paper is to present a complete tracking algorithm that can be applied to any image sensor. The IO-MeMBer filter is first used to generate point estimates of the multi-object state of the dynamical system. The likelihood has been specifically adapted to utilize prior road information to constrain estimates of objects to roads hence increasing the accuracy of the estimates. We finally use an efficient track management scheme parallel to the multi-Bernoulli filter to package the estimates into more useful track trajectories.

A. Road constraints

In certain situations such as aerial surveillance prior information on terrain can be utilized to refine tracking estimates [19]. In algorithms that analytically propagate the object statistics it can be difficult to incorporate constraints such as terrain and roads. A general constraint on an object's dynamics will typically result in a non closed form posterior hence requiring an approximation to incorporate the information.

By using a particle or Sequential Monte Carlo approach it is simple to incorporate road constraints into the algorithm [19], [20].

We adopt the road model from [21] where roads are described by a list of trapezoidal segments. Curved roads can be approximated by using a set of trapezoidal segments. Incorporating road constraints into our image observation filter is then accomplished by factoring the map into the likelihood. The likelihood of an object off-road is given a penalty based on its distance away from the road. The filter is initialized by setting births at each road entry and exit point to the region. This also provides a simple method for initializing the algorithm.

For simplicity our approach reduces the likelihood of off-road locations to zero, resulting in the multi-object likelihood

$$g\left(y|X\right) = K^{-1}\mathbf{1}_{\mathbf{S}}\left(y\right) f\left(y\right) \prod_{x\in X} g_y\left(x\right)$$

where $\mathbf{1}_{\mathbf{S}}\left(y\right)$ is an indicator function, $\mathbf{S}$ is the union of all trapezoidal segments approximating the road and K^{-1} is a normalization factor.

B. Track management

A simple method to add track labeling to the multi-Bernoulli filter is to add labels within the Bernoulli components depend each new Bernoulli component inherits the parent label [15]. This is simple to implement but is only effective when targets do not intersect or even become close to each other. If targets do interact they will inevitably come within the track merging threshold and the label structure is quickly invalidated. To solve this we adopt the method in [22].

The simple labeling scheme of Bernoulli components only allows for labels to be associated directly to the past time steps labels. By using a more advanced track management method like [22], previous labels over several time steps can be considered in the association.

The basic scheme of the track management based on [22] is to first search for possible associations between the current and previous time step estimates. Second, previously missed estimates are considered for association with current estimates not yet associated. The third step is to then validate the current estimate not yet associated as a newly born target which will be assigned a new label. Finally if old estimates are not associated a certain amount of times it is eliminated from the memory of associated estimates. A more detailed pseudo-code is given below in 1.

An operation was added to eliminate noise picked up as track estimates near the birth. This operation confirms a track after a threshold number of time steps. A check is used to ensure objects move a minimum distance so that zero velocity tracks which stay at the birth location fail to be confirmed. This ensures that the spurious pixels near the edges do not corrupt the track estimates.

IV. Numerical Simulation

This section will show the new tracker applied to a challenging road constrained track before detect scenario. Prior road information is available to the tracker is in figure 1. The sensor information is a sequence of two dimensional images that changes as time passes. The whole image is passed directly to the first stage of the tracker, the IO-MeMBer filter. The tracker uses the beginning and ends of the road map as birth locations for new targets that can appear within the scenario whilst producing point estimates as candidates for track components. The output from the IO-MeMBer filter is then passed to the track management routine to reconstruct the final track estimates.

Algorithm 1 Track management pseudo-code

INITIALIZATION

1) At the first time step
2) Acquire first set of estimates from IO-MeMBer filter
3) Assign association bits for each estimate to 1, missed estimate bits to 0
4) Assign new label per each estimate

ITERATIVE LABELING

1) At every progressive time step repeat
2) Compare current to directly previous estimates
    a) If within an association distance threshold
        i) Assign previous estimate label to new estimate and set corresponding association bit to 1
3) Compare current estimates to previous estimates that have been missed
    a) If within an association distance threshold (missed time $*$ threshold)
        i) Assign previous estimate label to new estimate and set corresponding association bit to 1
4) If estimate is not associated
    a) Assign new label and and set corresponding association bit to 1
5) If estimate in memory is not associated in the current time step
    a) Increase its missed estimate counter
    b) If counter reaches threshold remove estimate from memory

A. Observation Model

At each time increment, the observation is a two-dimensional image consisting of an array of cells with a scalar intensity. In our scenario, the observation region is a $4000m \times 4000m$ square region, and the observation image is a $500pix \times 500pix$ array, giving each cell side lengths of $\Delta_x = \Delta_y = 8m$. As the observation is a two dimensional image, it will be treated as an array with the index of ordered pair of integers $i = (a, b)$, where $1 \le a, b \le 500$. The model of the observation is given by the probability density of the intensity $y_{i,k}$ of pixel i, at time k, given a state x_k:

$$p(y_{i,k}|x_k) = \begin{cases} \mathcal{N}\left(y_{i,k}; h_{i,k}(x_k), \sigma^2\right), & i \in T\left(x\right) \\ \mathcal{N}\left(y_{i,k}; 0, \sigma^2\right), & i \notin T\left(x\right) \end{cases} \quad (2)$$

where $\sigma^2 = 1$ is the noise variance, $h_{i,k}$ is the point spread function given by

$$h_{i,k}(x_k) = \frac{\Delta_x \Delta_y I_k}{2\pi\sigma_h^2} \exp\left(-\frac{(\Delta_x a - p_{x,k})^2 + (\Delta_y b - p_{y,k})^2}{2\sigma_h^2}\right)$$

with source intensity $I_k = 1250$, blurring factor $\sigma_h^2 = 40$, and $(p_{x,k}, p_{y,k})$ being the position of the state x_k. The effective template $T(x_k)$ is the $4pix \times 4pix$ square region whose center is closest to the position of x_k. A typical observation is shown in Figure 2.

B. Constant Turn Dynamic Model

For our scenario a generic constant-turn model is adopted. A constant-turn model is expected to enable more accurate tracking on curved roads as well as to better accommodate sharp turns performed by the target. We hence model the target state at time k by a five dimensional vector $x_k = [\ p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k}, \omega_k]^T$, consisting of the x-coordinate, x-velocity, y-coordinate, y-velocity and turn rate, and whose dynamics is described by the following state transition matrix equation

$$x_k = \psi_k(x_{k-1}) + Gw_{k-1} \tag{3}$$

where

$$\psi_k(x_{k-1}) = \begin{bmatrix} 1 & \frac{\sin(\omega_{k-1}T)}{\omega_{k-1}} & 0 & \frac{\cos(\omega_{k-1}T)-1}{\omega_{k-1}} & 0 \\ 0 & \cos(\omega_{k-1}T) & 0 & -\sin(\omega_{k-1}T) & 0 \\ 0 & \frac{1-\cos(\omega_{k-1}T)}{\omega_{k-1}} & 1 & -\frac{\sin(\omega_{k-1}T)}{\omega_{k-1}} & 0 \\ 0 & \sin(\omega_{k-1}T) & 0 & \cos(\omega_{k-1}T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1}, \tag{4}$$

$$G = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}, \tag{5}$$

$T = 1s$ is the sampling period, $w_{k-1} = [w_{\dot{x},k-1}, w_{\dot{x},k-1}, w_{\omega,k-1}]^T$ is a vector of velocities and turn-rate noise components, which are zero-mean Gaussian with standard deviations $\sigma_{\dot{x}} = \sigma_{\dot{y}} = 5m/s$ and $\sigma_\omega = 6°/s$ respectively.

C. Actual scenario

In our scenario targets follow the roads shown in figure 1. There are 4 targets in the scenar io each which start at a different time. The scenario lasts for a duration of $K = 100s$. The exact start stop parameters and ground truths is shown in figure 3. A single frame of the sequence of image observations is also shown in figure 2.

D. Simulation parameters

The track management parameters used are maximum missed estimates is 7, track confirmation occurs after 7 successive estimates, tracks can be initiated $50m$ from birth locations and tracks are propagated only when the next estimate is under $100m$ away. The birth process is a multi-Bernoulli density which contains a Gaussian at each extreme of the roads. Hence there is 8 birth locations. Each is initialized with a covariance of $P_\gamma = \left[20, 20, 20, 20, \frac{6\pi}{180}\right]^T$.

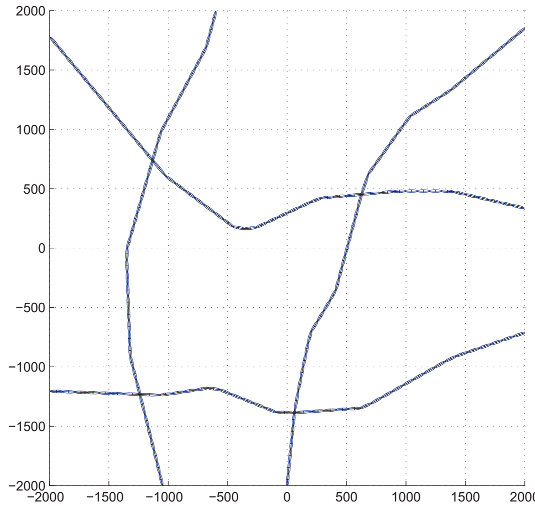

Fig. 1. Prior road information. Birth locations are at extremities of roads.

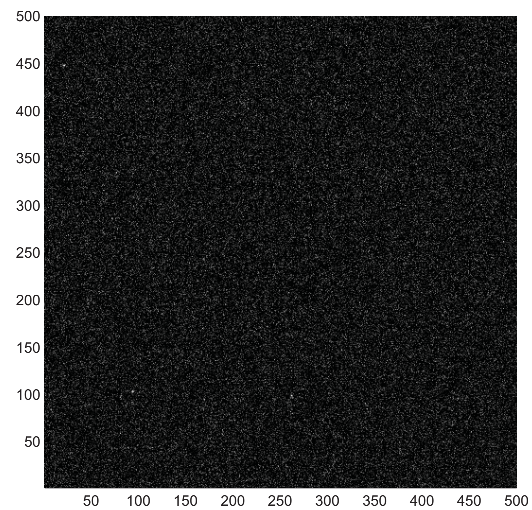

Fig. 2. A single image observation

E. Simulation results

The outputs of the tracker are shown in figures 4 and 5. The tracker has handled the scenario quite well and is able to identify all births and deaths. There are occasional incidence of dropped and false tracks.

We then use the a consistent multi-object tracking metric: the Optimal Sub-Pattern Assignment for tracks (OSPA-T) [23] to analyze the performance of the tracker. We use the parameters $c = 100$m, $p = 1$, $\Delta = 80$m and $\alpha = 25$m where c is the cutoff parameter which determines the relative penalty given to cardinality and localization errors, p represents the order of

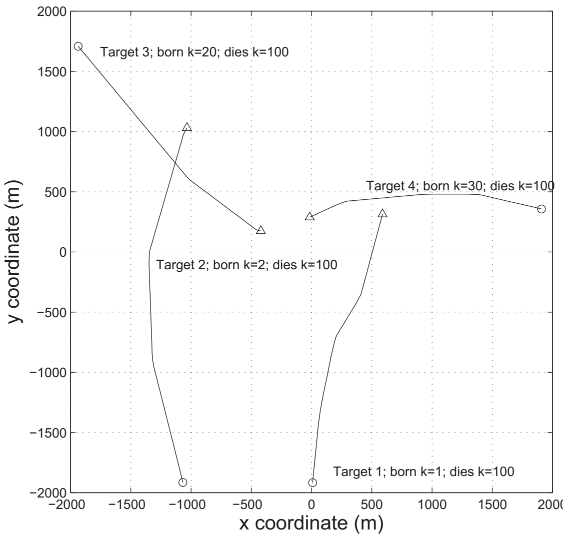

Fig. 3. Ground truths for scenario

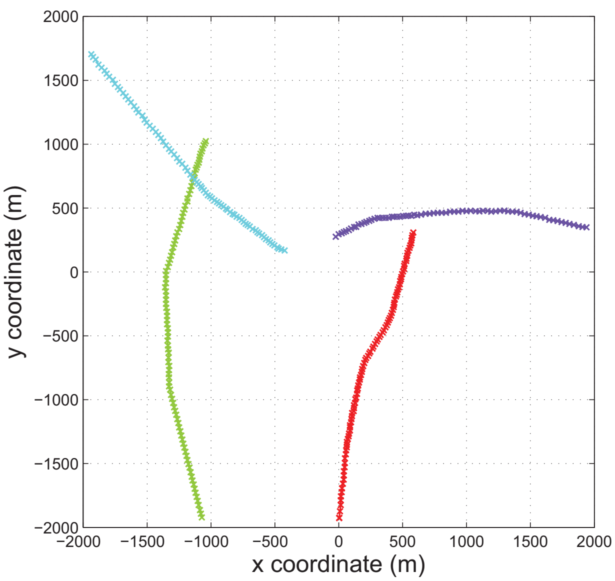

Fig. 4. Color coded track output

the per-target localization and cardinality error, Δ represents the penalty at a specific time if a track does not exist, and α is the penalty for any labelling errors. The total OSPA-T consists of cardinality and localization components. Full details of the OSPA-T can be found in [23]. The results shown in figure 6 indicate that tracker performance is acceptable. The spikes corresponds to noise initiating a track earlier than when it actually began. For the remainder of the scenario the OSPA-T remains reasonably low, with the small error most likely due to the pixelisation of physical locations due to the nature of image observations.

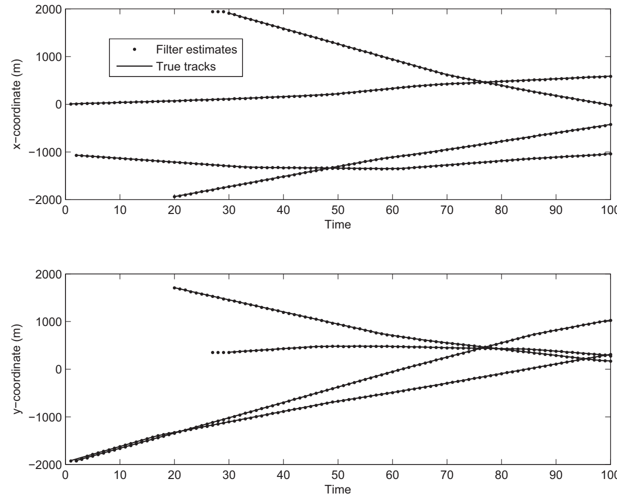

Fig. 5. Tracker estimates against time

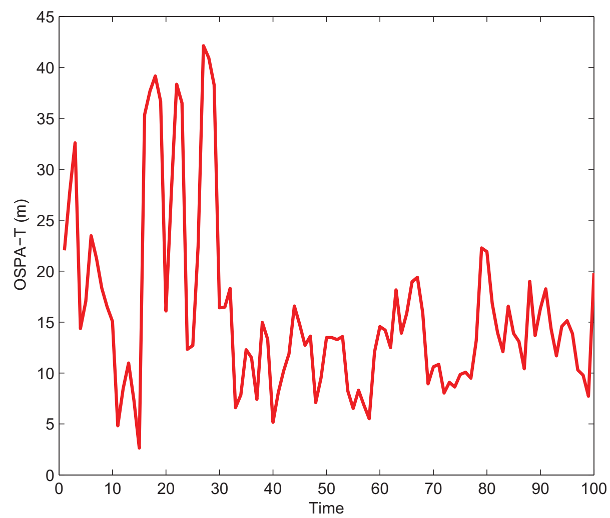

Fig. 6. OSPA-T distance for scenario ($c = 100$m, $p = 1$, $\Delta = 80$m, $\alpha = 25$m)

V. Conclusion

We have demonstrated that by using a combination of the IO-MeMBer filter with road constraints and track management an effective multi-object image observation tracker can be created. The numerical example demonstrates its effectiveness in a track before detect scenario. The tracker is naturally limited by some of the fundamental limitations of the original IO-MeMBer filter. Overlapping objects due to target intersections can still cause dropped estimates. Future work will examine ways of obtaining better initializations and accommodating target intersections within the framework of the IO-MeMBer filter.

Acknowledgment

Jeff Wong is supported by a DSTO Australia Postgraduate Top-Up Scholarship.

References

[1] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. San Diego, CA: Academic Press, 1988.

[2] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[3] D. J. Salmond and H. Birch, “A particle filter for track-before-detect,” *Proc. American Control Conference,* Vol. 5, pp. 3755-3760, Arlington, Va, USA, June 2001.

[4] J. Czyz, B. Ristic, B. Macq, A particle filter for joint detection and tracking of color objects, *Image and Vision Computing*, vol. 25, no. 8 ,pp. 1271–1281, 2007 .

[5] S. Buzzi, M. Lops, L. Venturino, M. Ferri, Track-before-detect procedures in a multi-target environment, *IEEE Trans. Aerosp. Electron. Syst.,* vol. 44, no. 3, pp. 1135–50, 2008.

[6] R. L. Streit, M. L. Graham, and M. J.Walsh, “Multitarget tracking of distributed targets using histogram-PMHT,” *Digital Signal Processing*, Vol. 12, No. 2-3, pp. 394–404, 2002.

[7] S. Davey, M. Rutten, and B. Cheung, “A comparison of detection performance for several Track-Before-Detect algorithms,” *EURASIP Jrn. Advances in Signal Processing*, Vol. 2008 , Issue 1, Article 41, 2008.

[8] R. Mahler, *Statistical Multisource-Multitarget Information Fusion.* Norwood, MA: Artech House, 2007.

[9] B.-T. Vo, and B.-N. Vo, “A Random Finite Set Conjugate Prior and Application to Multi-Target Tracking,” *Proc. IEEE Intl. Conf. Intelligent Sensors, Sensor Networks, and Information Processing*, Adelaide, Australia, Dec. 2011.

[10] R. Mahler, “Multi-target Bayes filtering via first-order multi-target moments,” *IEEE Trans. Aerosp. Electron. Syst*., vol. 39, no. 4, pp. 1152-1178, Oct. 2003.

[11] R. Mahler, “PHD filters of higher order in target number,” *IEEE Trans. Aerosp. Electron. Syst*., vol. 43, no. 3, Jul. 2007.

[12] B. N. Vo, S. Singh and A. Doucet, “Sequential Monte Carlo methods for Bayesian Multi-target filtering with Random Finite Sets,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224-1245, Oct. 2005.

[13] B.-N. Vo and W.-K. Ma, “The Gaussian mixture Probability Hypothesis Density filter,” *IEEE Trans. Signal Processing*, Vol. 54, No. 11, pp. 4091–4104, Nov. 2006.

[14] B.-T. Vo, B.-N. Vo, and A. Cantoni, “Analytic implementations of the Cardinalized Probability Hypothesis Density filter,” *IEEE Trans. Signal Processing*, Vol. 55, No. 7, pp. 3553–3567, Jul. 2007.

[15] B. T. Vo, B. N. Vo and A. Cantoni, “The Cardinality Balance Multi-Target Multi-Bernoulli Filter and Its Implementations,” *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409-422, Feb. 2009.

[16] B. N. Vo, B. T. Vo, N. T. Pham and D. Suter, “Joint Detection and Estimation of Multiple Objects from Image Observations,” *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5129-5241, Oct. 2010.

[17] R. Hoseinnezhad, B. N. Vo,D. Suter and B. T. Vo, “Multi-object Filtering from Image Sequence Without Detection,” *Proc. ICASSP 2010*, pp. 1154-1157, Mar. 2010.

[18] R. Hoseinnezhad, B. N. Vo, B. T. Vo and D. Suter, “Visual Tracking of Numerous Targets via Multi-Bernoulli Filitering of Image Data,” *Pattern Recognition* in press (accepted April 2012). **

[19] M. Ekman and E. Sviestins, “Multiple Model Algorithm Based on Particle Filters for Ground Target Tracking,” *Proc. of 7th Int. Conf. on Information Fusion*, 2007.

[20] B. Ristic, S. Arulampalam, and N. J. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Norwood, MA: Artech House, 2004.

[21] B. T. Vo, C. M. See, N. Ma and W. T. Ng, “Multi-sensor Joint Detection and Tracking with the Bernoulli Filter,” *IEEE Trans. Aerosp. Electron. Syst.,* vol. 48, no. 2, pp. 1385-1402, 2012.

[22] K. Shafique and M. Shah, “A Non-Iterative Greedy Algorithm for Multi-fram Point Correspondence,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 1, pp. 55-65, Jan. 2005.

[23] B. Ristic, B.-N. Vo D. Clark and B.-T. Vo, “A metric for Performance Evaluation of Multi-Target Tracking Algorithms,” *IEEE Trans. Signal Processing*, Vol. 59, no. 7, pp. 3452-3457, Jul. 2011.